Frequent Episode Mining to support Pattern Analysis in Developmental Biology

Bathoorn R., Welten M., Richardson M., Siebes A., Verbeek, F.J.

ronnie@cs.uu.nl ; fverbeek@liacs.nl

Imaging & BioInformatics, LIACS, Leiden University, the Netherlands (MW, MR, FJV)
Distributed Databases, Computer Science, Utrecht University, the Netherlands (RB, AS)

Keywords: frequent episode mining, heterochrony, pattern analysis, developmental biology

Abstract. We introduce a new method for the analysis of heterochrony in developmental biology. Our method is based on methods used in data mining and intelligent data analysis and applied in, e.g., shopping basket analysis, alarm network analysis and click stream analysis. We have transferred, so called, frequent episode mining to operate in the analysis of developmental timing of different (model) species. This is accomplished by extracting small temporal patterns, i.e. episodes, and subsequently comparing the species based on extracted patterns. The method allows relating the development of different species based on different types of data. In examples we show that the method can reconstruct a phylogenetic tree based on gene-expression data as well as using strict morphological characters. The method can deal with incomplete and/or missing data. Moreover, the method is flexible and not restricted to one particular type of data: i.e., our method allows comparison of species and genes as well as morphological characters based on developmental patterns by simply transposing the dataset accordingly. We illustrate a range of applications.

1 Introduction

The relation between evolution and development is intriguing [11,12] and considered essential for gaining understanding in the tree of life. Heterochrony, defined as the change of timing in events in development leading to changes in size and shape of species, facilitates analyzing differences in species. The key goal in heterochrony analysis is to relate evolutionary distance between species to changes in timing of developmental events. Tools to analyze developmental timing in a quantitative way, however, are not performing satisfactory; in particular for large datasets. In addition, for assessment, a relative timing is required and such is not present in existing computational

approaches. Therefore, complementary to published methods, such as event-pairing [11] and Search-based Character Optimization [15], we developed a method for heterochrony analysis that includes efficient extraction of developmental patterns and at the same time allows using different types of data, e.g. morphological and gene-expression, in a universal manner. To that end we propose an analysis of developmental sequences based on, so called, episodes [13]. Episodes are small, partially ordered, sets of events that frequently occur in the data. A collection of episodes extracted from a developmental sequence provides a good basis for further analysis of that sequence. For our method to run efficiently a special data structure is required to accomplish fast updates on the extracted patterns. We, therefore, propose a data structure referred to as the *episode tree* which is specifically designed for and tailored to this kind of application.
Our analysis starts with a dataset containing developmental sequences (cf. 2.3) and from this dataset an episode tree is created by sliding a window over the developmental sequences; all episodes found in this time window are added to the episode tree. Subsequently, a distance measure, based on the concept of *heterochrony*, is used between the entities in our dataset (species). After computation of the distances and clustering based on these distances, results are obtained and visualized as cladogram; typically showing evolutionary distance between species. Experiments with artificial, morphological and gene expression datasets are used to illustrate the scope of this method. In each of the experiments the entities we compare to each other can be different, i.e. clades, species or genes. Using a gene expression dataset as input results in a cladogram similar to those from biological literature. For our experiments we consider gene expression as extracted from patterns of gene expression from "*in situ*" hybridization, these are directly related to morphological characters. At this point, Micro Arrays gene expression patterns are not considered.

2 Materials & Methods

Here, we will describe the *Frequent Episodes mining in Developmental Analysis* (FEDA). The method is centered on a database contains the data to be analyzed as well as the patterns extracted. This approach facilitates the selection of interesting patterns for further analysis. Prior

to analysis the data are imported in the database using a comma separated values (CSV)-file (cf. Fig. 1A).

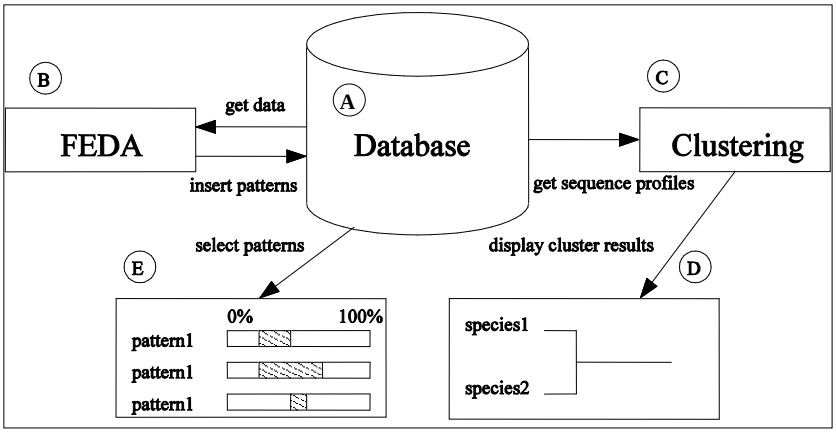

Fig. 1. Overview of FEDA architecture centered on a database. (A) Data import in the database. (B) FEDA finds frequent episodes and inserts these back in database. (C &D) Visual output, like clustering sequence profiles (C & D) or a pattern shift diagram (E) from frequent episodes.

2.1 Software and Hardware

The core algorithm is implemented in C++; additional data manipulation is accomplished by python scripting. For clustering the R statistical software [1] is used. Experiments were performed on a Desktop PC (1 GB RAM). For storage a MySQL database [2] is used.

2.2 Extraction of Episodes

We propose a method for finding sequence heterochrony in developmental sequences (cf. Fig. 1B). Using small frequently occurring patterns, called episodes, we try to find differences between developmental sequences. In order to have an unequivocal idea of the major concepts we define the core entities.

Definition 1 Developmental Sequence: A developmental sequence is an ordered list of developmental events describing the timing of these events within one species.

Definition 2 Episode: An episode [13] is a small ordered set of events that is frequent over all developmental sequences.

Definition 3 Frequency: The frequency of an episode is the total sum of the occurrences (cf. Def. 4) of this specific episode in all sequences.

For each occurrence its size is equal to or smaller than the maximum episode size.

Definition 4 Occurrence: For an episode to occur in a sequence, the events in this sequence need to be ordered strictly after each other in the same order as the events in the episode. Events in between that are not part of the episode may exist. Consequently, gaps between events in an episode can exist; events in an occurrence do not have to be contiguous.

Definition 5 Episode Size: The size of an episode occurring in a sequence ***s*** is the size of the smallest subsequence ***s'*** of our sequence in which the episode can still be matched. Such "match" is called an *occurrence* of the episode in ***s***. To limit the amount of episodes that can be found the size of the episodes has been restricted. The maximal episode size is the upper bound on the episode size.

The FEDA algorithm starts with a given maximal episode size and an empty episode tree as parameters. FEDA processes all the sequences in the data and integrates all occurrences of the episodes found in each sequence. This results in a collection of all episodes with the given maximal size together with their frequency. FEDA uses the episode tree to store this collection of episodes together with their frequency.

Definition 6 Episode Tree: An episode tree is a prefix-tree data structure on the episodes with the following features:

1. consists of nodes and children
2. the tree has an empty root node; the start of all the episodes
3. each node has zero or more child nodes and each node contains: *an event*, *a frequency* and *a binary list*
4. a node with no child nodes is called a leaf

The root node is the empty episode from which all other episodes are extended. All children of this root-node are episode trees containing episodes that start with the event contained in this node. In addition, each of the children stores a binary list with length equal to the number of sequences in the dataset and it holds a 1(*true*) at position 1 if the episode is found in the first sequence. This is the same for the other sequences in the dataset. In an episode tree the events found in all nodes passed in the path from the root to another node is an episode. An example of an episode tree is depicted in Figure 2.

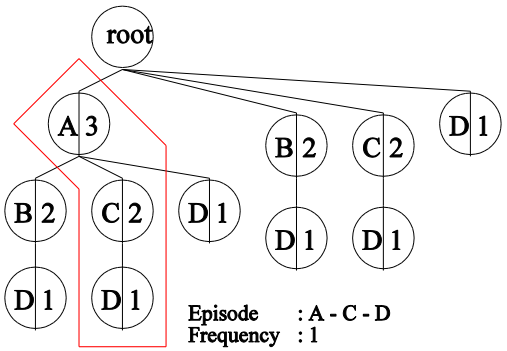

Fig. 2. An Episode Tree; the root is the start of all episodes contained in the tree. The highlighted path from the root to a leaf (end node) is the episode A-C-D with a frequency 1, as stored in the last node of the episode.

2.3 Episodes in Heterochrony analysis

After computation of all the frequent episodes it is exactly known which patterns occurred in which developmental sequence and the frequency of each pattern in all the developmental sequences. From the data sequence profiles are constructed; these are defined as:
Definition 7 Sequence Profile: A sequence profile is a vector that exactly shows which episodes where found in a given developmental sequence. The number of elements in this vector is equals the number of episodes found by FEDA. All episodes are indexed for the sequence profile. The value at each index is *true* if the episode was found or *false* if not found.
The sequence profiles can be used in standard clustering algorithms while still being able to capture the temporal dependencies in the developmental sequence. Furthermore, filters can be used to control the size of the profiles. A possible filter is to use all maximal frequent episodes instead of all frequent episodes and thereby choose a minimal frequency as a threshold.
Definition 8 Maximal Frequent Episodes: An episode is maximal frequent if it is not part of a larger frequent episode, i.e. a collection of maximal frequent episodes does not contain episodes that are part of other episodes in the collection.

2.4 Clustering of Sequence profiles

After the episode mining step, a sequence profile is obtained, indicating which episodes have been found in each of the sequences. This profile is used as a feature vector describing each of the sequences. To visualize the similarity/dissimilarity between sequences in the data, agglomerative clustering is applied on the sequence profiles (cf. Fig. 1C). To measure the distance between sequences a specific distance measure is required that excludes those episodes not present in both of the sequence profiles in the distance. The choice of the distance measure is motivated by the fact that an episode not being present in both sequence profiles is not contributing information on the biological difference between these two sequence profiles. This feature is typically expressed in the *Jaccard* distance [9], defined as:

$$Jaccard[i, j] = \frac{b + c}{a + b + c},$$

where both i and j are sequence profiles; a is the total number of episodes present in both i and j, b is the total number of episodes present only in i, c those only in j. In addition, d represents the episodes in none of the two profiles, d is not used in the computation but completes a cross table in the analysis of the profiles (cf. Fig. 3). It is easily seen that the *Jaccard* distance is a normalized figure; 0 for b,c=0 and 1 for a=0. In our analysis, the *Jaccard* distance reflects a relevance of the episodes found. Using the *Jaccard* distance a dissimilarity matrix is constructed by computing it for all possible pairs of two species (cf. Fig. 3); this matrix is used in the clustering. The agglomerative hierarchical clustering with complete linkage [10] is used; this is an unsupervised clustering method which initiates with a cluster for each of the entities present [1, *hclust*]. Subsequently, the two clusters that are closest are merged in a larger cluster and merging continues until all entities are in one cluster. The distance between two merged clusters is computed using complete linkage; i.e., all distances between all pairs of entities are computed and the largest of these distances is considered the distance between the two clusters. From the clustering outcome a cladogram can be derived (cf. Fig. 1D) visualizing the distances between all sequences in the dataset. The root of this cladogram is the point at which all species are joined in one large cluster whereas the leaves represent clusters containing only one species.

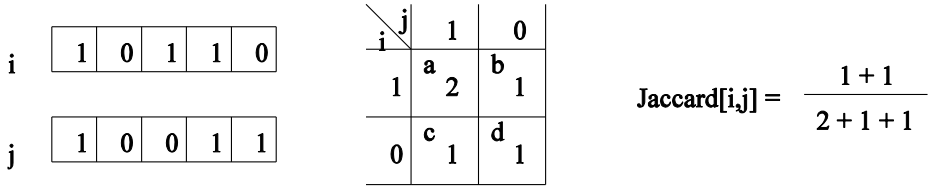

Fig. 3. Shown from left to right: the profiles *i* and *j*; a cross table recording the number of episodes shared by i and j (a) all episodes possessed by only one profile (b and c) and those contained in none of the profiles (d); the computation of the *Jaccard* distance.

3 Results

To show the different aspects of the method we present results of a number of experiments using three datasets: a small artificial dataset to demonstrate the method (cf. 3.1), a dataset of morphological events over time (cf. 3.2) and a gene expression dataset (cf. 3.4).

3.1 Artificial data

We will illustrate our method with a simple dataset consisting of 3 taxa and one outgroup [15]. It illustrates that FEDA treats the episodes as dependent features and as such is not prone to errors found in event pairing [11,12]. Feature dependency is preserved in the ordering of events in the episodes (cf. Table 1). We start with building a list of all episodes found in this dataset; we adhere to only adding episodes that are found in the data instead of all possible combinations of events that are found in our dataset. Table 2 contains a list of all the episodes that were found in each sequence resulting in a sequence profile for all 4 sequences. In Table 2 a "1" indicates that the episode was present in the sequence and a "0" indicates it was absent. Next, the dissimilarity matrix between all sequences is computed by summing all differences between each pair of profiles. For taxa 1 and 2 this results in 6 differences in their profiles (AB, AC, BA, CA, ABC, BCA). Repeating this for all sequences in the example results in a distance matrix (cf. Table 1B). This distance matrix shows that the distances between Taxa 1, 2 and 3 are all 0.86; the distance is computed using the *Jaccard* distance for all pairs of taxa. All pairs have 6 episodes for which the occurrence is different and 1 episode that is present in both taxa, resulting in a dissimilarity score of 6/7 between all the taxa, and a dissimilarity of 0 between the Outgroup and Taxon 1.

Finally, agglomerative clustering with complete linkage is applied, using the previously obtained distance matrix. The result is presented in Figure 4. The cladogram is realized by starting with all taxa in different clusters at the bottom of the cladogram and then merging clusters of the closest taxa. At completion, we end up with all taxa and the outgroup in one cluster at the top of the cladogram.

A	Sequence
Outgroup	ABC
Taxon 1	ABC
Taxon 2	BCA
Taxon 3	CAB

B	Out	T 1	T 2	T 3
Out	0			
T 1	0	0		
T 2	0.86	0.86	0	
T 3	0.86	0.86	0.86	0

Table 1. (**A**) Dataset of 3 taxons and 1 outgroup (**B**) Distance Matrix showing the distance between each pair of taxa based on Jaccard distance. All taxa are equally close to one another.

	AB	AC	BA	BC	CA	CB	ABC	BCA	CAB
Outgroup	1	1	0	1	0	0	1	0	0
Taxon 1	1	1	0	1	0	0	1	0	0
Taxon 2	0	0	1	1	1	0	0	1	0
Taxon 3	1	0	0	0	1	1	0	0	1
Totals	3	2	1	3	2	1	2	1	1

Table 2. Sequence Profiles recording the frequent episodes that occur in each taxon as well as the total number of times each episode occurs in the dataset.

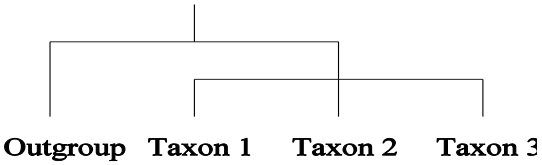

Fig. 4. The cladogram resulting from clustering the taxa based on the in Table 1B

3.2 Sequences of Morphological Characters in Development

Next, we illustrate FEDA with a more complex dataset [11] containing timed sequences of morphological events. Each event is taken from the development of one species. An event happens only once in a species. The dataset contains 14 entities (species), and one developmental sequence containing morphological events per entity (cf. Fig. 5).
If all frequent episodes were used in the clustering this would result in long runtimes and therefore the frequent episode set is reduced. Using only maximal frequent episodes (cf. Def. 8) in our experiments reduces the number of episodes in the clustering, as only the larger episodes are extracted. The window size was increased to obtain a sufficient number of features to cluster the data. For this particular dataset the parameters

for FEDA were set to a window size of 8 and a frequency threshold of 0.05, resulting in obtaining 983 episodes. In Figure 6 the resulting cladogram is depicted. The clustering is almost the same as the Taxonomy common tree [3], only minor differences are seen in the amphibians. The results obtained from event-pairing on this dataset [11] show the same pattern acknowledging that the granularity of the dataset is, actually, insufficient.

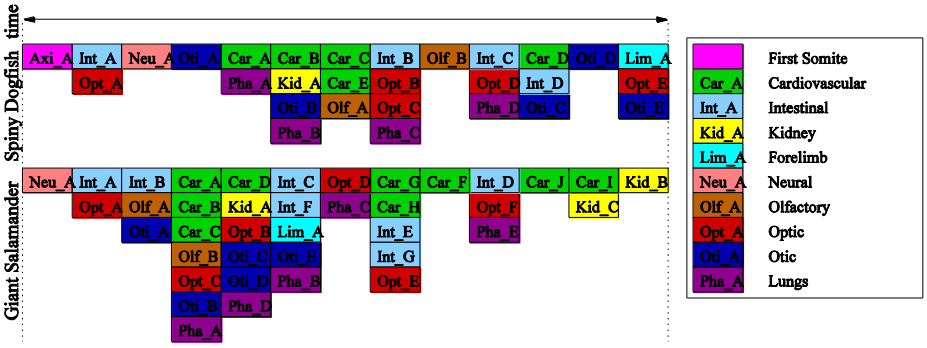

Fig. 5. Part of a recording of 2 developmental sequences presenting morphogical events over time. Here only spiny dogfish and giant salamander are depicted (dataset contains 14 species).

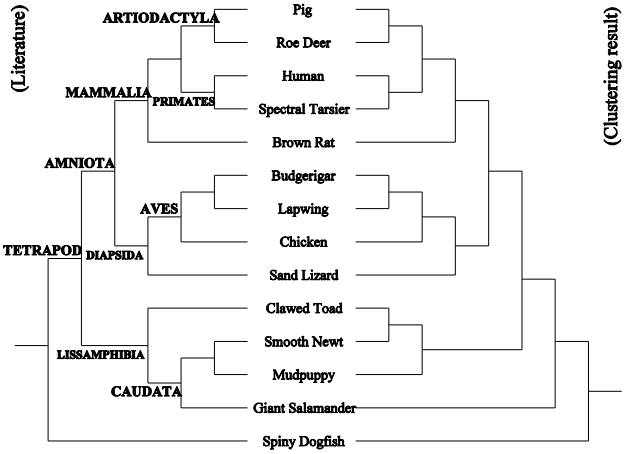

Fig. 6. Cladogram of the results computed by FEDA from a dataset of morphological events (right) compared to the taxonomy common tree from the NCBI (left).

3.3 Relative Timescale in Development

To allow linking patterns between different species, a relative timescale is introduced and used in the computations. This timescale is based on percentage of development of the species under study [23] and events

are linked relative to the developmental scale this species. E.g. gene *tbx5* is active in Zebrafish in [5% - 10%] of development (cf. Fig. 7).

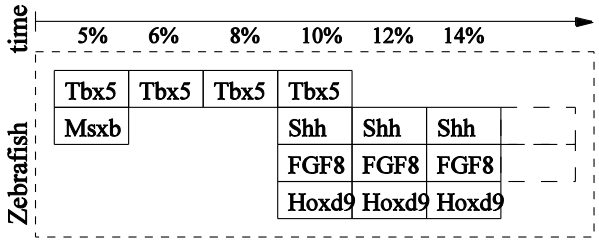

Fig. 7. Data recording of a selection of gene expression patterns in zebrafish in a relative time scale: *tbx5*, *msxb*, *ssh*, *fgf8*, *hoxb9*. At 10% of development 4 of the genes are expressed whereas at 14% development only 3 of the genes are expressed.

3.4 Sequences of Gene Expression in Development

Next, FEDA is applied to patterns of genes expression as found in the development of several model species. The clustering was performed with a window size of 4 and a minimal frequency of 0.04; the result corresponds with consensus in biological literature [14]. This result indicates that there is sufficient information in the data to differentiate between groups of species. The gene expression is analyzed to clades and therefore, subsequently, visualized as a cladogram. This cladogram is depicted in Figure 8.

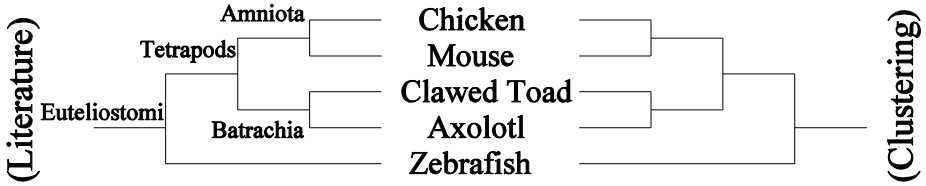

Fig. 8. Cladogram computed by FEDA based on gene-expression data (right) compared to a phylogenetic tree taken from biological literature [14] (left).

4 Conclusions and Discussion

We presented a method for the discovery of frequent patterns in a group of developmental sequences for quantitative analysis of heterochrony. All the episodes found in developmental sequences are found together with their frequency and a list of supporting sequences. These episodes are used in further analysis, such as clustering. Compared to previous experiments [4] our method is considerably

more efficient. Furthermore, we demonstrated that transpositions of the data enable comparing morphological characters and genes as well as species in a transparent way. We have illustrated that our algorithm works with artificial as well as biological data

Currently, two methods are used for the analysis of developmental sequences of events, i.e. Event-pairing [18,11,16,17,8] and Search-Based Character Optimization [15]. Over Event-pairing [11] our method has two advantages. It uses the data to determine which pairs are the most interesting to use and the "event-pairs" can contain more than two events, so, in fact they are groups of developmental events that co-occur frequently. Groups of events found by FEDA contain more information about developmental sequences compared to event-pairs.

Search-Based Character Optimization [15] shows excellent clustering results and can possibly also be applied in the analysis of gene expression data. Over this method FEDA has two advantages. It allows insight in clustering, because FEDA is based on frequent developmental patterns and these patterns can later on be used to obtain more insight into which patterns cause the tree to branch. In addition, FEDA scales better to the size of the dataset and the number of events used in the analysis. FEDA is only based on patterns that are frequent, thus allowing it to handle large amount of data with a large number of developmental events. This does not restrict our method to sequences that contain all events because in sequences with missing events we are still able to find developmental patterns, just not the patterns that contain this missing event. Furthermore, our method does not use an edit cost matrix. For long developmental sequences with a large number of possible events this edit cost matrix extends to enormous and impractical proportions. Our method has the advantage that no step costs have to be determined, because the distances between species are only based on the data and the developmental patterns found.

The episode tree mining algorithm (FEDA) is developed to scale to larger datasets that will become available from high throughput genomics and data from developmental biology [5,6,7,19,20,21].

5 Acknowledgements

This work is partially supported the Netherlands' council for Scientific Research (NWO), grant #050.50.213 (Bio Molecular Informatics).

6 References

[1] http://www.r-project.org
[2] http://www.mysql.com
[3] http://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi
[4] Bathoorn, R. and Siebes, A. 2004. Constructing (Almost) Phylogenetic Trees from Developmental Sequences Data. 8th European Conf on Principles and Practice of Knowledge Discovery in Databases: 500-502.
[5] Belmamoune, M., Verbeek, F.J. 2006. Heterogeneous Information Systems: bridging the gap of time and space. Management and retrieval of spatio-temporal Gene Expression data. InScit2006, Merida, Spain.
[6] Belmamoune M. and Verbeek, F.J. 2008 Data Integration for Spatio-Temporal Patterns of Gene Expression of Zebrafish development: the GEMS database. J.of Integrative BioInformatics, 5(2):92.
[7] Belmamoune, M., Potikanond, D., Verbeek, F.J. 2010 Mining and analysing spatio-temporal patterns of gene expression in an integrative database framework. J.of Integrative Bioinformatics, 7(3):128, 1-10.
[8] Bininda-Emonds, O.R.P, Jefferey, J.E., and Richardson, M.K. 2003. Is sequence heterochrony an important evolutionary mechanism in mammals? J. of mammalian evolution 10 (4):335-361.
[9] Jaccard, P. 1908. Nouvelles recherches sur la distribution florale. Bull Soc Vaudoise Sci Nat 44:223-227.
[10] Johnson, S.C. 1967. Hierarchical Clustering Schemes. Psychometrika 2:241-254
[11] Jeffery, J.E., Bininda-Emonds O.R.P., Coates M.I., Richardson, M.K. 2002. Analyzing evolutionary patterns in amniote embryonic development. Evolution & Development Volume 4 Number 4: 292-302.
[12] Jeffery, J.E., Richardson, M.K., Coates M.I., Bininda-Emonds, O.R.P. 2002. Analyzing Developmental Sequences within a Phylogenetic Framework. Systematic Biology Volume 51 Number 3: 478-491.
[13] Mannila, H., Toivonen, H. and Verkamo, A.I. 1995. Discovering frequent episodes in sequences. 1st Int. Conf. on Knowledge Discovery and Data Mining: 210-215.
[14] Metscher, B.D. and Ahlberg, P.E. 1999. Zebrafish in Context: Use of a Laboratory Model in Comparative Studies. Develomental Biology 210: 1-14
[15] Schulmeister, S. and Wheeler W.C. 2004. Comparative and Phylogenetic analysis of developmental sequences. Evolution & Development Volume 6 Number 1: 50-57.
[16] Smith, K.K. 2002. Sequence heterochrony and the evolution of development. Journal of morphology. 252:82-97.
[17] Smith, K.K 2003. Time's arrow: heterochrony and the evolution of development. Int. J. Dev. Biol. 47:613-621
[18] Schlosser, G. 2001. Using heterochrony plots to detect the dissociated coevolution of characters. Journal of experimental zoology (mol dev evol). 291:282-304.
[19] Verbeek, F.J., Lawson, K.A., and Bard, J.B.L. 1999. Developmental BioInformatics: linking genetic data to virtual embryos. Int.J.Dev.Biol. 43, 761-771.
[20] Verbeek,F.J., Rodrigues, D.D., Spaink H., Siebes A. 2004. Data submission of 3D image sets to a bio-molecular database using active shape models and a 3D reference model for projection. In: Proceedings SPIE 5304, Internet Imaging V. 13-23.
[21] Welten, M.C.M. 2007. Spatio-temporal gene expression analysis from 3D *in situ* hybridisation images. PhD Thesis, Leiden University.